University of Essex
Department of Mathematical Sciences

MA981: DISSERTATION

Predictive Modeling for Real Estate Prices Using Machine Learning: A Multi-City Analysis with Model Deployment

Musaddik Maulavi
2311452

Supervisor: Lan Truong

November 21, 2024
Colchester

Contents

List of Figures

List of Tables

CHAPTER 1

Abstract

This study focuses on predicting house prices using advanced machine learning techniques. The real estate market is complex and traditional methods are often inaccurate. This research aims to create a precise prediction model by collecting and analyzing data from various cities, identifying important features, and testing different machine learning algorithms using Hyper parameter tuning and comparing the models using the plots. The best model will be integrated into a user-friendly web application for buyers, sellers, investors, and other end users. This website is then hosted live on internet. This project covers cities like Bangalore, London, Perth, Pune, Kuala Lumpur, Lisbon, Cairo, Dubai, and Bangkok, providing a broad understanding of housing prices in different markets. This research offers significant benefits by providing accurate price estimates and enhancing the understanding of housing markets across different cities.

CHAPTER 2

Introduction

2.1 Background of Study

The real estate market has grown a lot recently, and knowing the prices of houses has become very important for people who want to buy, sell, or invest in homes. Predicting house prices accurately helps buyers make better choices, helps sellers set fair prices, and helps investors find good deals. In the past, experts and statistics were used to guess house prices. But now, with machine learning, we can make better and more accurate predictions. Machine learning models can find patterns in large amounts of data and use this to predict future prices [3].

2.2 Problem Statement

Predicting house prices accurately has always been challenging because real estate market is complex and always changing. Old traditional methods often don't work well because they use limited data and simple models, they ignore many factors that affect property prices. This can lead to wrong valuations, causing financial risk/issues for buyers, sellers, and other parties.

With the rise of machine learning and its algorithm, there is now a better way to tackle this problem. By using a large amount of data and advanced machine learning algorithms, We can now better understand the real estate market and improve the accuracy of house price prediction. However, there are many current prediction models that still don't fully take

advantage of machine learning and hyper parameter tuning. This research aims to fill that gap by creating a new, more advanced prediction model that uses the full power of machine learning technique.

Furthermore, to make these predictions accessible to end-users, I have developed a web application. This application allows users to input parameters affecting house prices and obtain accurate price predictions based on the advanced model.

2.3 Research Objectives

The main motive of this research is to create a precise and accurate model for predicting house prices using advanced machine learning techniques. This big goal is divided into several smaller tasks as listed and described below:

1. Collecting and Preparing Data: First, I will gather data from several cities from kaggle. This data will include several crucial information about houses such as past information, location and area in square feet/meters.

2. Creating useful features: I will identify and create important features that affect house prices. This is a key step to ensure the model understands the complexities of the real estate market. It involves turning raw data into meaningful features to train the model more efficiently.

3. Developing a model: I will use various machine learning algorithms, such as Linear Regression, Decision Trees, Lasso Regression, Gradient Boosting, SVR, and Decision Tree to build and compare prediction models by implementing Hyperparameter tuning as well. The aim is to find the algorithm that predicts house prices most accurately and reliably by utilizing hyper-parameter tuning.

4. Evaluating and Improving the Model: I will assess how well the different models perform using suitable metrics with the help of hyperparameter tuning. I will also plot the comparative plots. Based on this, I will optimize the best model to enhance its accuracy and ability to generalize.

5. Deploying and Testing the Models: The last step is to launch the best model in a web application using pickle. This website will allow users to enter property details and get price estimates, serving as a useful tool for different people in the real estate market.

2.4 Significance of Study

This research is very important for many people involved in real estate. For people looking to buy or sell homes, a good prediction model can help them make better decisions. Real estate agents and brokers can give more accurate price estimates, making their clients happier. Investors can use this model to anticipate how much money they might make from property investments. Policymakers can understand market trends better and create better rules. This study also helps academics by improving the ways to predict house prices.

2.5 Scope of the Study

This study looks at house prices in several cities: Bangalore, London, Perth, Pune, Kuala Lumpur, Lisbon, Cairo, Dubai, and Bangkok. These cities were picked to show different real estate markets. The aim of this study is to predict the house prices in these cities using past data from these cities. It does not include commercial properties or rental prices. And utilizing various machine learning algorithms for prediction. For the model building it is considering factors like location, property size, number of bedrooms and bathrooms. The best model is used to predict the price in the web application.

2.6 Research Questions

To guide the research process and ensure a thorough investigation, the following questions were created:

1. How can we use various machine learning to make home price predictions more accurate?

2. Which machine learning methods work best for predicting home prices in various cities?

3. How can end users effectively interact with and utilize the machine learning model through a web-based application to obtain accurate and user-friendly home price predictions? Specifically, what user interface features and input options will make the model accessible to a wide range of users, including those with limited technical knowledge?

4. How can the model's outputs be presented in a way that is easy to interpret and useful for decision-making?

2.7 Methodology Overview

The study starts with collecting data for several cities from a trusted website (For this project I have used Kaggle). This data will be cleaned to fix any missing or inconsistent information. If required new variables will be created to better understand the real estate market. Different machine learning algorithms will be tested, hyper-parameter tuning will also utilized to find the best model, and their accuracies can be measured using Mean Absolute Error (MSE) and Root Mean Squared Error (RMSE). The most accurate model will be fine-tuned and made available in an easy to use front end website.

2.8 My Contribution

In this research, I have made important contributions to understanding and predicting housing prices in three different cities: London, Bangalore, Cairo, Bangkok, Perth, Dubai, Pune, Kuala Lumpur, and Lisbon. Here is a summary of my contributions:

2.8.1 Data Integration and Preparation

- I gathered from various sources to create a complete dataset for each city.
- I created a strong data processing system that cleans the data, handles missing values, converts inconsistent formats, and normalizes the features. This makes sure the data is clean and ready for analysis, which is vital for building accurate models.

2.8.2 Advanced Feature Engineering

- I added new and useful features, like price per square foot and encoded location data, which improved the models' ability to make accurate predictions.
- I standardized measurement units across different datasets, making it easier to compare and train models accurately.

- I also converted the local price of that city in GBP Pound. So that the prices can be compared among the cities.

2.8.3 Exploratory Data Analysis (EDA)

- I performed thorough EDA, creating visualizations and statistical summaries to understand the data's patterns and trends.

- I visualized the top 10 site locations of each city with the highest house prices. And the chart is displayed on the website for each city.

- EDA helped me find and remove outliers, ensuring the models were trained on reliable and representative data.

2.8.4 Model Development and Evaluation

I tested various machine learning algorithms, including linear regression, ridge regression, lasso regression, and decision tree regression, to find the best models for predicting house prices.

- I fine-tuned the models' parameters and used cross-validation for Linear Regression as a base model to make sure the models work well with new, unseen data.

- I utilized the power of Hyper-parameter Tuning for algorithms such as Linear Regression, Lasso, Decision Tree, Random Forest, SVR, and Gradient Boosting to find out the best model that can be used for model deployment.

2.8.5 User-Centric Application

- I developed an easy-to-use web application that allows users to predict house prices by entering details like location, total square feet, number of bedrooms, and number of bathrooms.

- The app uses the best model from the hyper-parameter tuning to provide instant price predictions, making it a valuable tool for end users.

2.8.6 Comprehensive Analysis Across Multiple Cities

- Unlike many studies that focus on just one city, my research compares nine different cities. This gives a broader view of the factors affecting housing prices in different markets. This project will also allow end users to use the front-end web application.

2.9 Summary

The introduction chapter sets up the research by giving an overview of the topic, stating the problem, objectives, importance, methods, scope and my contribution to this topic. The next chapters will explore the existing literature, detailed methods, and results. This will help to understand how advanced machine learning techniques can predict home prices.

CHAPTER 3

Literature Review

Predicting house prices is an important research area with big impacts on the economy, real estate market, and personal finances. Accurate predictions can help investors and home buyers to make better decisions. In recent years, new machine learning techniques have greatly improved the accuracy and reliability of these predictions. This literature review looks at the latest advancements in house price predictions using machine learning, focusing on studies from the last 5-6 years.

3.1 Housing prices prediction with deep learning: an application for the real estate market in Taiwan : By Zhan et al. (2020) [37]

This literature review examines the paper "Housing Prices Prediction with Deep Learning: An Application for the Real Estate Market in Taiwan" by Zhan et al. The study focuses on predicting housing prices in Taiwan using advanced deep learning techniques, specifically the Back Propagation Neural Network (BPNN) and Convolutional Neural Network (CNN). This review will summarize the research, compare it with existing literature, critique its methodologies, and discuss its implications.[37]

The study by Zhan et al. created a dataset with information about houses and the economy from January 2013 to December 2018. The dataset includes details about housing transactions

(both "land + building" and "land + building + park") and various economic factors such as the investment demand ratio, owner-occupier ratio, price-to-income ratio, loan burden ratio, and bargaining space ratio. The authors used BPNN and CNN to build models to predict housing prices. They measured how well these models worked using metrics like Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), R-squared (R^2), and adjusted R-squared [37].

The study found that the CNN model, especially when it included specific housing features, provided the most accurate predictions. The results indicate that deep learning methods like CNN can significantly improve the accuracy of housing price predictions compared to traditional statistical models [37].

Zhan et al. concluded that deep learning models, particularly CNN, are highly effective for predicting housing prices. These models can be valuable for creating targeted strategies and interventions in the housing market [37].

3.2 Research on the Design and Application of House Price Prediction Algorithms and Model Based on Machine Learning: By Kexin Chen and Jianhui Huang (2023) [4]

This paper by Kexin and Jianhui presents a study focused on utilizing advanced machine learning algorithms to improve the accuracy of housing price predictions [4].

The study uses a detailed dataset and follows several steps to analyze the data. First, the data is visualized to understand its patterns. Then, it is cleaned and prepared for analysis. After that, important features (attributes) are created to help improve the models. Finally, various ensemble models, which are combinations of different machine learning algorithms, are implemented and evaluated based on how well they predict housing prices [4].

The results show that ensemble models perform much better at predicting housing prices compared to single models. The study points out that cleaning the data and creating useful features are crucial steps that significantly improve the accuracy of the models. Additionally, it was found that ensemble models can handle complex relationships between different factors more effectively [4].

The paper concludes that using machine learning, especially ensemble models, is a strong

approach for predicting housing prices. These models are able to provide more accurate predictions than traditional methods. This makes them highly valuable for people involved in the real estate market, such as buyers, sellers, and investors. The study also suggests that further improvements in data processing techniques can lead to even better performance of these models [4].

3.3 A Reinforcement learning-based weight fusion algorithm for House Price Prediction: By Zhang, Fan, and Gou (2023) [38]

This paper by Zhang, Fan, and Gou (2023) introduces an innovative model that uses reinforcement learning to enhance prediction accuracy by dynamically adjusting the weights of various prediction models [38].

This paper tackles the challenge of accurately predicting house prices using different types of data. It introduces a new model that combines the strengths of multiple prediction algorithms through a method that uses reinforcement learning to adjust their weights [38].

- **Feature Engineering:** The dataset includes both categorical and continuous variables. Categorical variables are transformed using methods like ordinal, one-hot, and label encoding. Continuous variables are processed by examining their Pearson's correlation with house prices [38].

- **Prediction Models:** Several models are used for prediction, including multiple linear regression, regression decision tree, extreme random regression tree, and multilayer perceptron. The weights of these models are adjusted using a reinforcement learning technique called Q-Learning to minimize the root mean square error (RMSE) [38].

- **Reinforcement Learning:** The Q-Learning algorithm dynamically updates the weights of the models to improve prediction accuracy [38].

The proposed weight fusion algorithm significantly improves the accuracy of house price predictions compared to using individual models alone. On the Shenzhen dataset, the

improvements in RMSE range from 8.65% to 64.61%. On the Boston dataset, the improvements range from 53.06% to 76.40%. This indicates that combining multiple models using reinforcement learning can lead to much better predictions [38].

The reinforcement learning-based weight fusion algorithm effectively enhances the accuracy of house price predictions. This method shows great potential for broader applications in real estate forecasting and could be used in other areas where accurate predictions are crucial. By leveraging the strengths of different models and dynamically adjusting their contributions, this approach provides a more robust and reliable prediction tool [38].

3.4 House Price Prediction Approach Based on Deep Learning and ARIMA Model: By Feng Wang, Yang Zou, Haoyu Zhang, and Haodong Shi [30]

This study addresses the challenges of predicting house prices by using a combined approach of deep learning and the ARIMA model. This combination aims to improve accuracy and handle large datasets effectively.[30]

The researchers used TensorFlow to implement their model, training it with a dataset of 11,937 samples taken from a Chinese property website. The model includes an input layer with 13 features, four hidden layers, and an output layer. They used the Adam optimizer and the ReLU activation function for training the deep learning model. To predict house price trends, they used the ARIMA model, which incorporates historical price data to forecast future trends.[30]

The proposed model showed better performance compared to the Support Vector Regression (SVR) model. It achieved lower root mean square error (RMSE) and mean relative error (MRE) on both training and test datasets. The deep learning model was more accurate in predicting individual house prices, while the ARIMA model was effective in forecasting overall price trends. This dual approach ensured precise and reliable predictions.[30]

The combination of deep learning and ARIMA models significantly improves the accuracy of house price predictions. This method is effective in predicting both individual house prices and overall price trends. By using both models, the researchers were able to handle the complexity of the data and provide more accurate predictions. This approach can be useful

for real estate forecasting and other areas where accurate predictions are essential. The success of this model demonstrates the potential of combining different techniques to tackle challenging prediction problems.[30]

3.5 House Price Prediction Using Linear and Lasso Regression: By Mayank Sharma, Rahul Chauhan, Swati Devliyal, and Kanegonda Ravi Chythanya (2024) [27]

This paper aims to find ways to improve the accuracy and efficiency of predicting house prices using machine learning algorithms. The authors use three different regression techniques: linear regression, lasso regression, and ridge regression [27].

- **Data**: The study uses a dataset from Bangalore, which includes variables such as the number of bedrooms, bathrooms, the area of the house, and its location [27].
- **Regression Techniques**: The three methods applied are:
  - **Linear Regression**: A basic technique that finds the best-fitting line through the data [27].
  - **Lasso Regression**: Similar to linear regression but adds a penalty to reduce the number of variables [27].
  - **Ridge Regression**: Also similar to linear regression but adds a penalty to shrink the coefficients of the variables [27].

The study finds that both linear and ridge regression models provide more accurate predictions than the lasso regression model. The linear and ridge models result in lower error percentages, meaning they predict house prices more accurately [27].

The final models developed using linear and ridge regression are practical tools for real estate stakeholders because they offer better accuracy in predicting house prices. These findings suggest that for datasets like the one from Bangalore, ridge regression and linear regression are more reliable choices compared to lasso regression. Additionally, the improved accuracy of these models can help real estate professionals make better-informed decisions, benefiting buyers, sellers, and investors [27].

3.6 Housing Price Prediction Model Using Machine Learning: By Aman Chaurasia and Iman Ul Haq (2023) [3]

This study proposes a model that implements various ML techniques to predict housing prices with improved accuracy. The main question this paper addresses is how well machine learning algorithms can predict housing prices using different features from real estate data [3].

- **Data and Techniques:** The authors use a mix of data preprocessing techniques and machine learning algorithms, including linear regression [3].
- **Dataset**: The data includes various features like property characteristics, neighborhood demographics, and economic indicators [3].
- **Training and Testing:** The data is divided into training and test sets to evaluate the model [3].
- **Evaluation Metrics:** The linear regression model is assessed using Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) [3].

The linear regression model shows a strong ability to predict housing prices. The MAE value is USD 82,288.22, and the average predicted price error is +- 6.67 %. The model also successfully identifies the most important factors that affect housing prices [3].

The study concludes that machine learning algorithms, especially linear regression, can accurately and reliably predict housing prices. This can help various people in the real estate market, such as buyers, sellers, and investors, make better decisions. Additionally, using these algorithms can streamline the process of assessing property values, potentially saving time and resources for real estate professionals. The authors suggest that future research could explore combining linear regression with other machine learning methods to further enhance prediction accuracy [3].

3.7 Improved Prediction Accuracy of House Price Using Decision Tree Algorithm over Linear Regression Algorithm: By Pammi Chandu and N. Bharatha Devi (2023) [2]

This study aims to enhance the accuracy of house price predictions by using a Decision Tree (DT) algorithm instead of the traditional Linear Regression (LR) method [2]. The study investigates whether the Decision Tree algorithm can provide more accurate house price predictions compared to the Linear Regression algorithm [2].

- **Sample Size:** The study uses a sample size of 20, with 10 participants in each group (one group for Decision Tree and one group for Linear Regression) [2].
- **Data Source:** The dataset was obtained from Kaggle and includes various attributes that affect house prices [2].
- **Algorithms Compared:** The study compares two algorithms: Linear Regression (Group 1) and Decision Tree (Group 2) [2].
- **Statistical Tools:** IBM SPSS was used for statistical analysis. An independent sample T-test was conducted to compare the accuracies of the two algorithms [2].
- **Accuracy**: The Decision Tree algorithm achieved a mean accuracy of 90%, while the Linear Regression algorithm achieved 80% [2].
- Statistical Significance: The statistical analysis showed that the difference in accuracy between the two algorithms was not significant ($p=0.618$, $p>0.05$) [2].

The study concludes that the Decision Tree algorithm performs better than the Linear Regression algorithm in predicting house prices, with higher accuracy. However, the difference in accuracy between the two algorithms is not statistically significant. This means that, while the Decision Tree algorithm appears to be better, the improvement might not be reliable across different samples. Therefore, both algorithms could still be useful, depending on the specific context and dataset. Future studies could include larger sample sizes and additional algorithms to explore further improvements in prediction accuracy [2].

3.8 Machine Learning based House Price Prediction Model: By Chen Chee Kin, Zailan Arabee Bin Abdul Salam, and Kadhar Batcha Nowshath (2022) [13]

Chen Chee Kin, Zailan Arabee Bin Abdul Salam and Kadhar Batcha Nowshath from the Asia Pacific University of Technology Innovation, explores the use of machine learning (ML), artificial neural networks (ANN), and chatbot technologies to predict house prices in Malaysia [13].

This study aims to develop a model to predict house prices effectively using machine learning techniques to aid buyers and sellers in making informed decisions [13].

The paper uses different machine learning algorithms to build models that predict house prices. These algorithms include:

- **Linear Regression**
- **Decision Tree Regressor**
- **Artificial Neural Networks**

The data for the study was collected through online questionnaires because of COVID-19 pandemic restrictions [13].

The study discovered that the decision tree algorithm predicted house prices more accurately than the linear regression model. The decision tree model was better at handling non-linear relationships and interactions between different features, making its predictions more accurate and reliable [13].

The decision tree algorithm is found to be a better choice for predicting house prices compared to linear regression. This is because it provides higher accuracy and can handle complex datasets more effectively. Given its robustness and ability to deal with non-linear relationships, the decision tree model can be a valuable tool for real estate professionals. In the future, combining decision trees with other advanced techniques, like ensemble methods or neural networks, could further improve prediction accuracy and provide even more reliable tools for the real estate market [13].

3.9 Modeling House Price Prediction Model using XG Boost and Machine Learning Algorithms: By Ajmeera Kiran et al. (2023) [14]

Ajmeera Kiran and others explores the application of machine learning algorithms to predict house prices, emphasizing the use of XGBoost along with other regression models [14].

This paper addresses the challenge of predicting house prices by leveraging advanced machine learning techniques. The primary research question is: "How can machine learning algorithms, particularly XGBoost, improve the accuracy of house price predictions compared to traditional methods?" [14].

- **Data Collection:** The study utilizes a comprehensive dataset encompassing various features related to houses, such as location, size, age, number of rooms, and proximity to amenities. The dataset was carefully curated to ensure it captures the key variables that influence house prices [14].

- **Models Used:** The research compares the performance of several machine learning algorithms, including Linear Regression, Decision Trees, Random Forest, and XGBoost. These models were selected for their varied approaches to handling data and capturing relationships between features [14].

- **Evaluation Metrics:** The models' prediction accuracies were assessed using metrics such as Mean Squared Error (MSE) and R-squared. These metrics provide insights into the models' precision and their ability to explain the variance in house prices [14].

- **Performance**: The XGBoost model significantly outperformed the other algorithms, delivering the lowest MSE and the highest R-squared values. This indicates that XGBoost provided more accurate and reliable predictions compared to traditional methods [14].

- **Significance of Features:** The analysis revealed that features related to the location and structural attributes of the house were the most significant predictors of house prices. This highlights the importance of these factors in the valuation process [14].

- **Effectiveness of XGBoost:** The paper concludes that XGBoost is exceptionally effective for house price prediction due to its ability to handle complex interactions between features and deliver high accuracy. XGBoost's gradient boosting framework allows it to model non-linear relationships and capture intricate patterns in the data [14].

- **Future Work:** The authors suggest further research into hybrid models that combine the strengths of multiple algorithms to potentially enhance prediction accuracy further. Additionally, they recommend incorporating more diverse datasets, including data from different regions and market conditions, to improve the model's generalizability and robustness [14].

3.10 Optimization Techniques for Deep Learning Based House Price Prediction: By Dr. J. Vijaya et. al (2023) [8]

This paper explores advanced optimization methods to enhance the accuracy of house price predictions using deep learning models [8].

The main question this paper explores is: "How can different optimization techniques make deep learning models better at predicting house prices?" [8]

The authors used a detailed method that included cleaning and preparing data, picking the right features, and fine-tuning the settings of a Deep Neural Network (DNN). They used several optimization techniques, including Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), and Grey Wolf Optimizer (GWO), to improve the performance of the model [8].

The experiments showed that combining Genetic Algorithm with Ant Colony Optimization (GA-ACO) and using Mayfly-Wolf Optimization (MF-WO) produced the best results. The GA-ACO technique achieved a Mean Absolute Error (MAE) of 0.0765 and an accuracy of 99.7%, while the MF-WO method resulted in an MAE of 0.092 and an accuracy of 99.2% [8].

The paper concludes that using advanced optimization techniques greatly enhances the accuracy of deep learning models in predicting house prices. These techniques make the models more reliable and precise, which can help in making better forecasts in the real estate market [8].

3.11 Research on the application of integrated RG-LSTM model in house price prediction: By Guang Wang and Zubao Shu (2023) [31]

This study proposes a novel approach integrating regression analysis with long short-term memory (LSTM) networks for predicting house prices.

The main question this paper explores is: "How can combining regression analysis with LSTM (Long Short-Term Memory) models improve the accuracy and reliability of predicting house prices?" [31]

The authors used several steps to conduct their research:

- **Creating a Housing Price Index**: They reviewed existing studies to establish a comprehensive system for tracking housing prices [31].
- **Applying the RG-LSTM Model**: They applied a combined model of regression analysis and LSTM (RG-LSTM) to predict housing prices in Nanjing, China [31].
- **Comparative Experiments**: They compared the performance of the RG-LSTM model with traditional time series prediction models [31].

The study found that the RG-LSTM model provided more accurate and reliable predictions for house prices compared to traditional methods. This model effectively captured the complex, nonlinear relationships in the housing data [31].

The integrated RG-LSTM model proved to be a better choice for predicting house prices because it handles the nonlinear components and relationships in the data more effectively than traditional models. This approach can be particularly useful for anyone involved in the real estate market, such as investors, developers, and policymakers, who need accurate predictions to make informed decisions [31].

3.12 Stacking Ensemble Learning for Housing Price Prediction: a Case Study in Thailand: By Gan Srirutchataboon et al (2021) [28]

This paper proposes a stacking ensemble learning framework to predict housing prices in Thailand, combining convolutional neural networks (CNN) with various ensemble models [28].

The main question this paper addresses is: "How can using stacking ensemble learning improve the accuracy of predicting house prices in Thailand? [28]"

The study used data from a major Thai real estate website and Open Street Maps (OSM). This data included details about houses and geographic information. The researchers developed a model that uses a Convolutional Neural Network (CNN) to extract features from house images. They combined this with different ensemble methods like random forests (RF), extreme gradient boosting (XGBoost), and adaptive boosting (AdaBoost). Finally, they used a linear regression model to fine-tune the predictions [28].

The study discovered that the stacking ensemble model performed much better than individual models. Specifically, the model that combined CNN and XGBoost had the best results, with a Mean Absolute Percentage Error (MAPE) of 17.83%. This was significantly lower than the MAPE values of models that didn't use CNN [28].

The study by Srirutchataboon et al. (2021) shows that using a stacking ensemble learning framework significantly improves the accuracy of predicting house prices in Thailand. By combining Convolutional Neural Networks (CNN) with ensemble models like Random Forest (RF), Extreme Gradient Boosting (XGBoost), and Adaptive Boosting (AdaBoost), the researchers developed a highly effective prediction model. The model that integrated CNN and XGBoost performed the best, achieving a Mean Absolute Percentage Error (MAPE) of 17.83%, which was notably better than models that did not use CNN. This demonstrates the effectiveness of stacking ensemble learning in capturing complex patterns and improving prediction accuracy.

CHAPTER 4

Methodology

4.1 Introduction

The methodology chapter explains the steps taken to carry out this research. This study aims to create an accurate and reliable model for predicting house prices using advanced machine learning techniques and to build a web application for amazing user experience by just putting the parameters such as location, number of rooms and bathrooms. To do this, I followed a structure approach that includes research design, data collection, data analysis, model building and using pickle to export the model to the web application.

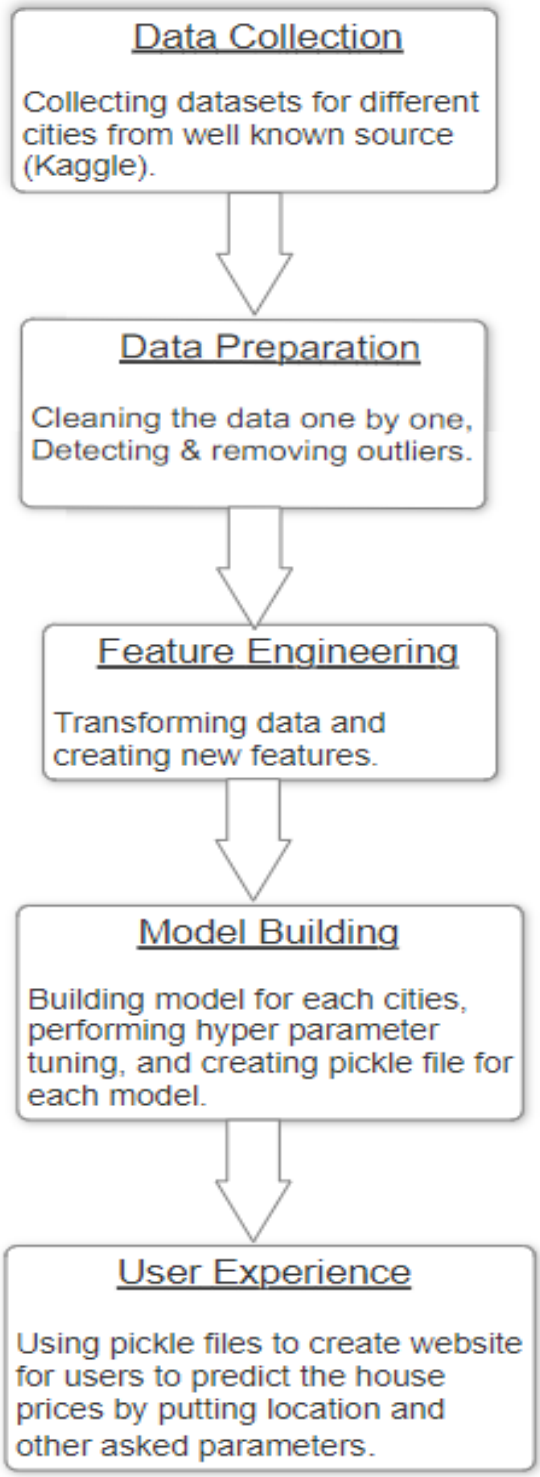

Figure 4.1: Process Flowchart

The above flowchart shows the structured process I followed to create an accurate home price prediction model. I started with data collection, gathering datasets for different cities from a well-known source, Kaggle. Next, I moved on to data preparation, where I cleaned

the data one by one, detecting and removing any outliers. After that, I performed feature engineering, transforming the data and creating new features to improve the model's accuracy. In the model building phase, I built models for each city, performed hyper parameter tuning, and created pickle files for each model. Finally, I focused on the user experience by using these pickle files to create a website where users can predict house prices by entering the location and other relevant parameters. This methodical approach ensures that our prediction model is both reliable and user-friendly.

This chapter clearly explains each part, making it easy to understand and use for similar projects.

4.2 Introduction to Machine Learning

This section is all about machine learning and related topics, explaining how to choose the right model and measure the performance of machine learning algorithms. This section also gives brief history of machine learning, highlighting its strengths, challenges and opportunities [1].

So, What is Machine Learning?

In the year 1959, an American Computer scientist Arthur Samuel coined the term 'Machine Learning'. He defined it as a machine's ability to learn without being explicitly programmed. In other words, machine learning algorithms analyze input data to predict output values. They improve their performance as they receive more data, and they gain intelligence over time. Mostly, 'Machine Learning' and 'Artificial Intelligence' are used interchangeably, but in reality machine learning is actually a subset of AI [1].

How Machine Learning Works?

Basically, machine learning has the ability to identify and predict patterns in data without prior knowledge of the data. These methods rely on mathematics, statistics, and computer science. Whereas traditional algorithms, machine learning techniques are designed to learn from the data and adjust model parameters and variable weights without human intervention [1].

Types of Machine Learning?

There are many machine learning algorithms that have been developed in the past few decades. And it is important to classify them due to their diversity. There are three primary

types of machine learning: supervised learning, unsupervised learning, and reinforcement learning [1].

1. Supervised Learning:

In supervised learning, input and output data both are provided to the algorithm. Learning involved adjusting the algorithm's parameters and the weights of the input variables. Regression and classification are common methods used in supervised learning. Regression models use continuous input variables to predict the numerical output, while classification models label input variables [1].

2. Unsupervised Learning:

This type of machine learning algorithm is useful for discovering hidden patterns or grouping data into clusters. Unsupervised learning addresses problems with insufficient information. These algorithms do not have any prior knowledge of the outcome unlike Supervised learning [1].

3. Reinforcement Learning:

This type of machine learning algorithm works on trial and error basis. This algorithm focuses on learning through positive and negative reinforcement. Behaviours that achieve the desired outcome are rewarded, while those that do not are penalized [1].

Why Machine Learning Matters?

By continuously learning from the past data, these analytic methods can even help determine the best strategies to achieve desired results. This capability of machine learning makes it powerful tool in various fields, from health care and finance to transportation and entertainment. [1]

4.3 Data Collection

The dataset used for this research is collected from the publicly available source i.e Kaggle. I focused on 9 cities: Bangalore, Pune, Kuala Lumpur, London, Bangkok, Cairo, Perth, Dubai and Lisbon. These dataset contain information about property prices, sizes, locations and other important details which includes Number of Bedrooms and Bathrooms.

Brief description of each dataset:

Bangalore Dataset: This dataset is taken from kaggle (https://www.kaggle.com/code/cheshta09/ba
house-prices/input). It includes details like area type, site location, balcony, total square feet

and local price which is in Indian Rupees (INR).

Pune Dataset: This dataset is taken from same source (Kaggle: https://www.kaggle.com/code/mahr houses-predection/input). And it has same column names as of Bangalore Dataset.

Cairo Dataset: This dataset is also downloaded from same source (Kaggle: https://www.kaggle.com/ housing-prices-eda-model-comparative-study/input?select=properties.csv). This dataset has columns like house type, price in EGP, number of bedrooms, Area in square metres bathrooms, house title and location.

Bangkok Dataset: This dataset is has attributes like Property type, Property area in sq. ft, Bathrooms, Bedrooms and Price in local currency i,e THB (https://www.kaggle.com/datasets/varintorn housing-condo-apartment-prices).

London Dataset: This dataset has columns like: Property name, Site location, Number of Bedrooms, Bathrooms, Number of Reception (Irrelevant Feature), price in local currency i.e GBP. Also this dataset is collected from same source (Kaggle: https://www.kaggle.com/datasets/arnavk prices-in-london).

Perth Dataset: This dataset is downloaded from Kaggle (https://www.kaggle.com/code/cristhianco house-price-predictions). This dataset includes attributes like address, suburb, price, bedrooms, bathrooms, garage, landArea, floorArea, buildYear, cbdDist, nearestStn, nearestStnDist, dateSold, postcode, latitude, longitude, nearestSch, nearestSchDist, nearestSchRank.

Dubai: Again the dataset is collected from Kaggle (https://www.kaggle.com/code/mohammedobeid prices-linear-model/input). This dataset includes feature such as Number of bedrooms, bathroom, price in AED, Site location and property area.

Kuala Lumpur Dataset: This data is gathered from Kaggle again. It includes size, bedrooms, bathrooms, price in RM, and location.

Lisbon Dataset: Collected from Kaggle again (https://www.kaggle.com/code/cgrodrigues/lisbon-house-prices-with-regression/input). It includes property subtype, bedrooms, bathrooms, area in square meters, district, parish, and price in Euros.

4.4 Data Preprocessing

Data preprocessing involves several key steps to clean and transform the raw data into the suitable format for analysis and modeling.

4.4.1 Handling Missing Values:

- Missing values were identified and addressed. Missing values in several datasets were removed by removing whole row. As the handling missing values is really crucial in model prediction. It causes less accuracy and uncertain prediction.

4.4.2 Feature Engineering:

- New features were created with an existing one to enhance the datasets. PriceGBP was derived from the local currency with accurate conversion rate for all the dataset.

- The property size of few datasets were in square metres. It is converted into square feet by multiplying it by conversion rate. So that property area is in same unit for all the datasets.

4.4.3 Location Normalization:

- Locations with low occurrence were grouped into a single category 'other' to simplify the analysis.[26, 15]

4.5 Exploratory Data Analysis

EDA was performed to gain the insights into the data from the listed nine cities and understand the underlying patterns and distributions.

4.5.1 Visualization:

- Scatter Plots: These were used to see the relationship between total square feet and price in different locations. This helps in understanding how price varies with size.

- Bar Plots: These plots were utilized for most of the cities to analyze the distribution of bathrooms, compare average prices by property type, and highlight the top 10 locations with the highest house prices.

4.5.2 Outlier Detection and Removal:

- **Removing Outliers Based on Price Per Square Foot (PPS)**: This method eliminates properties whose price per square foot is too high or too low compared to the average price in their location, ensuring only typical values are considered.

- **Removing Outliers Based on Bedrooms (BHK):** This method filters out properties whose price per square foot is significantly lower than properties with one less bedroom in the same location, provided there are enough properties in the comparison group to make a reliable judgment.

- Once identified, these outliers were removed to improve the accuracy of our predictive models. [33, 34]

4.6 Feature Engineering

Feature Engineering is nothing but creating new features that could potentially enhance the model's performance. In simpler terms, it's like preparing ingredients for a recipe [17, 10, 40, 7].

4.6.1 Price per Square Feet:

This feature was calculated for almost all the dataset to normalize the price data. It helps in comparing properties of different size on a common scale.

4.6.2 Conversion of Units:

- For Datasets such as Dubai, Cairo, Lisbon, Kuala Lumpur and Perth which has property area in square metres is converted into square feet by multiplying it by conversion factor. This conversion ensured consistency across all datasets, making it easier to compare and analyze data from different cities.

- For all the datasets, the house prices which were originally in local currency were converted into GBP to compare the prices, as it is easy to compare when the columns are in same unit.

4.6.3 Location Encoding:

It is really important to convert the categorical data into numbers before using it in machine learning model. So for this Locations were converted from categorical data into numerical format using one-hot encoding. This method creates binary columns for each location, which makes the data suitable for machine learning.

All these steps in EDA and feature engineering are really important in transforming the raw data into a more structured and meaningful format, making it easier to analyze and build accurate predictive models. Valuable insights can be gained with the help of visualization and statistical analysis. Whereas feature engineering helped in creating consistent and useful features for modeling. [39, 16]

4.7 Model Building

Several machine learning models are utilized to predict house prices of several cities. These models include Linear Regression, Ridge Regression, Lasso Regression and Decision Tree Regressor.

4.7.1 Model Selection:

1. Multiple Linear Regression:

    - This is a basic regression model. It helps us create a simple line that best fits the data points. I used this model as a starting point as to know how good my data is for model prediction.
    - Multiple regression is a statistical method used to understand the relationship between one dependent variable and multiple independent variables [24]. It can be used for tasks like describing data, making predictions, and estimating control [29]. The basic model includes a dependent variable Y and several independent variables $x_1, x_2, \ldots, x_k$. This relationship is represented by the equation [24]:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \epsilon_i \tag{4.7.1}$$

        – y_i: Value of the dependent variable for the i-th observation.

- x_{ij}: Value of the j-th independent variable for the i-th observation.
- β_0: Y-intercept of the regression surface.
- β_j: Slope of the regression surface with respect to x_j.
- ϵ_i: Random error for the i-th observation.

In this model, there are n observations and k predictors, where n must be greater than $k + 1$.

2. Ridge Regression:

- The ridge regression method reduces errors by making coefficients closer to zero. This is done by minimizing the following equation [36]:

$$\hat{\beta}^R = \sum_{i=1}^{k} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \tag{4.7.2}$$

where:

- $\hat{\beta}^R$ = estimated coefficients in the ridge regression model
- k = total number of observations
- y_i = value of the dependent variable for the i-th observation
- β_0 = intercept term
- p = total number of independent variables
- x_{ij} = value of the j-th independent variable for the i-th observation
- β_j = coefficient for the j-th independent variable

The estimated coefficients are found using:

$$\hat{\beta}^R = (X^T X + cI)^{-1} X^T Y \tag{4.7.3}$$

where X is the matrix of independent variables, Y is the dependent variable, I is the identity matrix, and c is the ridge parameter [36].

3. Lasso Regression:

- LASSO regression is a method that helps improve a model by making some feature values smaller or even zero if they don't help the model. It does this by adding a penalty for using too many features, so only the most important ones are kept [5].

This model also builds on Linear Regression but uses L1 regularization, where as Ridge uses L2. L1 regularization can help with feature selection by shrinking less important feature to zero. This makes it useful for datasets with many features or sparse data. It help to increase model accuracy.

4. Decision Tree Regressor:

   - Decision Trees (DTs) [21] are a popular method for classification. They work by creating a tree from training examples, where each node represents an attribute, and each branch represents a possible value of that attribute. To classify a new instance, you start at the root of the tree, test the attribute values, and follow the branches that match the instance's attribute values until you reach a leaf node, which gives the class label [5].

5. Gradient Boosting:

   - The Gradient Boosting Regressor (GBR) [25] is an ensemble model that combines several weak models, typically decision trees, to create a stronger model. Each new model in the sequence learns from the errors of the previous model. This process, called "boosting," improves the accuracy of the predictions. The GBR uses gradient descent to minimize errors by adjusting the initial predictions with new ones, resulting in a final model that is a weighted combination of all the individual models [25].

6. Support Vector Regressor:

   - Support Vector Regression (SVR) [5] is an extension of Support Vector Machine (SVM) that aims to predict continuous values. It introduces a region called the tube around the function to minimize prediction errors. SVR uses an epsilon-insensitive loss function, which only penalizes predictions that are more than epsilon away from the true value. This means that only significant errors are penalized. The width of the tube is determined by the value of epsilon, and different loss functions like linear or quadratic can be used [5].

7. Random Forest:

- Random Forest is a machine learning method used for classification and regression. It creates a "forest" of many decision trees, each working with different parts of the original data to reduce overfitting. Each tree makes its own prediction, and in Random Forest Regression, the final prediction is the average of all these individual tree predictions [12].

4.7.2 Model Training and Evaluation

1. **Data Splitting:**

   - Datasets are splitted into training and testing in ratio 4:1 i.e 80 percent and 20 percent respectively. The training set is used to train the model, and the testing set is used to evaluate its performance.

2. **Model Evaluate Metrics:**

   - **R-squared (R^2) :** The coefficient of determination, or R^2, tells us how well a model fits the data. In simple terms, it shows how close the regression line is to the actual data points. A higher R^2 means better performance [23].
   - **Mean Absolute Error (MAE):** Mean Absolute Error (MAE) is a common metric used to measure prediction accuracy. Unlike Root Mean Squared Error (RMSE), which gives more weight to larger errors, MAE treats all errors equally. MAE is straightforward because it increases linearly with the size of errors, making it easy to understand. It calculates the average of the absolute differences between predicted and actual values, ensuring all errors are positive. Lower MAE value indicates better accuracy [19].
   - **Mean Squared Error (MSE):** Mean Squared Error (MSE) measures how well a model's predictions match actual values. It does this by averaging the squared differences between observed and predicted values. Squaring the differences ensures that positive and negative differences don't cancel each other out, giving a clearer picture of prediction accuracy. Lower MSE means better performance [20].

3. **Cross-Validation:**

- ShuffleSplit is practiced for cross validation. Cross-validation is a method to test how well a machine learning model works with new data. It involves splitting the data into several parts. One part is used for testing the model, while the rest are used for training. This process is repeated multiple times with different parts used for testing each time. The average of these tests gives a better idea of the modelâs overall performance [6].

Hyperparameter Tuning

1. **GridSearchCV:**

   - This is a technique to find the best combination of hyperparameters for a model. In this research I used **Linear, Lasso, Ridge Regression, Gradient Boosting, SVR and Decision Tree Regressor** to improve their performance by testing different parameter values.

Model Evaluation

1. **Visual Assessment:**

   - Residual plots are generated to show the difference between actual and predicted values. Along with scatter plots to compare the actual and predicted values to visually assess how well the models performed.

2. **Comparison:**

   - The evaluation metrics such as R^2, MAE and MSE for each model is calculated and compared them to determine which model performed best.

By following all the steps above, I ensured that our models were trained, evaluated, and fine-tuned effectively to provide accurate predictions for housing prices. [11, 9, 22]

4.8 Comparative Analysis

The average housing prices across the selected nine cities were compared to understand the variations in real estate markets globally.

1. **Mean Price Calculation:** All the local prices were converted into the GBP. Then mean price in GBP is calculated for each city.

2. **Visualization:** A bar plot was created to compare the average housing prices across different cities.

3. **Statistical Analysis:** In order to determine if the differences in housing prices were significant, Statistical tests were performed.

4. **Model Evaluation Comparison:** The outcome of Hyper-parameter tuning is compared for each city.

4.9 Model Deployment

This is another important but complex part of our study. For practical application for the users, the best-performing model for each city was saved using pickle, and this pickle was used while creating a website. Not only this, the feature column for all the cities were saved in a JSON file.

4.9.1 Web Application Development

I developed a web application using the Flask framework in Python, which serves as the backend of the application. Flask is a lightweight yet powerful framework used for building web applications.

Backend Development

1. **Flask Setup:**

    - Flask handles HTTP requests from clients (web browsers) and interacts with the trained machine learning model to make predictions based on user inputs.

2. **Model Loading:**

    - I loaded a pre-trained machine learning model using pickle. This model was trained earlier to predict home prices based on factors like area, number of bedrooms, and bathrooms.

3. **Prediction Endpoint:**

    - A route (/predict home price) was defined to accept POST requests with property details (city, location, area sqft, bedrooms, bathrooms). Upon receiving these inputs, Flask uses the model to predict the home price and returns the result in JSON format.

Frontend Development

The frontend of the application was built using HTML, CSS, and JavaScript to create a user-friendly interface.

1. **User Input Form:**

    - HTML forms allowed users to input details such as city, location, area in square feet, number of bedrooms, and number of bathrooms.

2. **JavaScript for Dynamic Updates**

    - Javascript functions are implemented to handle the user interaction. Like suppose, if the user changes the city from the drop down it will automatically update the map and as well as the site location drop down.

Integration of Leaflet.js for Maps

1. **Map Integration**

    - Leaflet.js is used to create and manage interactive maps. Each city had predefined coordinates, and selecting a city dynamically updated the map to show the selected location.

User Interface and Interaction

The user interface (UI) is designed to be user-friendly.

1. **UI Components:**

    - Users can select a city from a dropdown menu, choose a specific location within that city, input property details, and click a "Predict Price" button to get predictions.

2. **Dynamic Updates:**

    - Predicted prices and map views updated dynamically based on user inputs, providing an interactive experience.

This web application allows users to predict home prices based on various factors and visualize property locations on an interactive map, enhancing usability and accessibility for real estate analysis and decision-making.

4.10 Tools and Libraries

Several tools and libraries are used in this research:

1. **Python:** The main programming language I used for data analysis and building models.
2. **Pandas:** A library used for data manipulation and analysis, making it easier to work with large datasets.
3. **NumPy:** A library for numerical operations, helpful in performing mathematical calculations.
4. **Matplotlib and Seaborn:** Libraries used for creating visualizations like bar plots and scatter plots.
5. **Scikit-learn:** A library used for machine learning algorithms and model evaluation.
6. **GridSearchCV:** A tool used to find the best parameters for our models, improving their performance.
7. **pickle:** A module used to save the trained models.
8. **JSON:** A format for saving the feature columns, making it easy to share and reuse the data structure.

4.11 Conclusion

This approach explains how in this study I analyzed house prices and developed predictive models for nine cities (Pune, Bangalore, London, Kuala Lumpur, Bangkok, Dubai, Perth,

Lisbon and Cairo). I began with data preprocessing, which involves cleaning and organizing raw data to make it useful for analysis. This process includes handling missing values, converting features into usable formats, and removing inconsistencies. Next, I performed EDA to understand the patterns and characteristics of the data. This involved generating statistical summaries and creating visualizations to see relationships and distributions within the data. I then implemented feature engineering, where new features were created that could improve the models performance. For example, I calculated price per square foot, converted measurement units, and encoded locations into numerical values. And lastly, machine learning models were used that learn from data to make accurate predictions, to build models that can predict house prices based on the features I engineered, The insights and models developed from this study can help stakeholders in the real estate industry make better decisions and analyze the market more effectively. By following this approach, I can better understand housing prices and predict them more accurately for different cities.

CHAPTER 5

Results

In this dissertation, I aimed to predict house prices in various cities using different machine learning models. I focused on thorough data preparation, feature creation, model testing to gain important insights and model deploying to achieve good results.

5.1 Data and Feature Engineering Outcomes

5.1.1 Price per Square Foot Calculation:

- I normalized the price data by calculating the price per square foot. This made it easier to compare properties of different sizes. This step was crucial for standardizing the data across different cities, making our analysis more reliable and comparable.

5.1.2 Conversion of Units:

- For cities like Dubai, Cairo, Lisbon, Kuala Lumpur, and Perth, I converted property areas from square meters to square feet. This ensured consistency in data representation, so all properties were measured on the same scale.

5.1.3 Location Encoding:

- I transformed categorical location data into numerical data using one-hot encoding. This method created binary columns for each location, making the data suitable for

machine learning algorithms and improving the model's ability to understand location information effectively.

5.2 Model Performance and Evaluation

This section summarizes how different machine learning models performed in predicting house prices in nine cities: Bangalore, Pune, Cairo, London, Bangkok, Dubai, Perth, Kuala Lumpur, and Lisbon. I used R-squared (R^2) values to measure how well each model predicted house prices. Higher (R^2) values mean better predictions [41, 18, 32, 35].

5.2.1 Model Comparison Across Cities

Linear Regression:

- Performed well in most cities, with R^2 values from 0.653859 (Lisbon) to 0.987885 (Bangkok).
- Best results were in Bangkok (0.987885) and Pune (0.85387), showing a strong relationship between the features and house prices in these cities [41, 18, 32, 35].

Lasso Regression:

- Similar to Linear Regression with slight improvements in some cities.
- Notable (R^2) values include 0.987888 (Bangkok) and 0.859097 (Pune), showing that L1 regularization helps improve model accuracy [41, 18, 32, 35].

Decision Tree Regressor:

- Varied performance with (R^2) values from 0.553138 (London) to 0.996913 (Bangkok).
- Excellent results in Bangkok (0.996913) and Cairo (0.860984304), showing its ability to handle complex relationships [41, 18, 32, 35].

Random Forest:

- Consistently high (R^2) values, with top performances in Bangkok (0.995817) and Cairo (0.910964955).
- Effective in reducing overfitting, as seen in Perth (0.875662) and Dubai (0.875662) [41, 18, 32, 35].

Gradient Boosting:

- High performance, similar to Random Forest, with (R^2) values like 0.995664 (Bangkok) and 0.868879518 (Cairo).
- Improved robustness and accuracy, especially in London (0.787324) and Dubai (0.841105) [41, 18, 32, 35].

Support Vector Regressor (SVR):

- Lower (R^2) values compared to other models, except in Pune (0.853559) and Lisbon (0.645117).
- Poor performance in Cairo (-1.04039) and Perth (0.249942), indicating difficulty handling complex data in these cities.

5.3 Key Observations

1. **Best Performing Models:**
    - **Decision Tree Regressor** and **Gradient Boosting** achieved the highest (R^2) values, particularly in Bangkok and Cairo. These models are effective for complex datasets [41, 18, 32, 35].
2. **City-Specific Performance:**
    - Bangkok had the highest model performance across various algorithms, suggesting a clear relationship between features and house prices [41, 18, 32, 35].

- Cairo showed significant variability, with some models like **Decision Tree Regressor** and **Random Forest** performing very well, while SVR struggled [41, 18, 32, 35].

3. **Model Robustness:**
   - **Random Forest** and **Gradient Boosting** demonstrated strong robustness and generalization across multiple cities, making them reliable choices for predicting house prices in different locations [41, 18, 32, 35].
4. **Some Limitations:**
   - The Perth dataset is so large and complex that models like Decision Trees, Random Forests, and Gradient Boosting are almost impossible to use. These algorithms would require so many resources and so much time that implementing them isn't feasible.

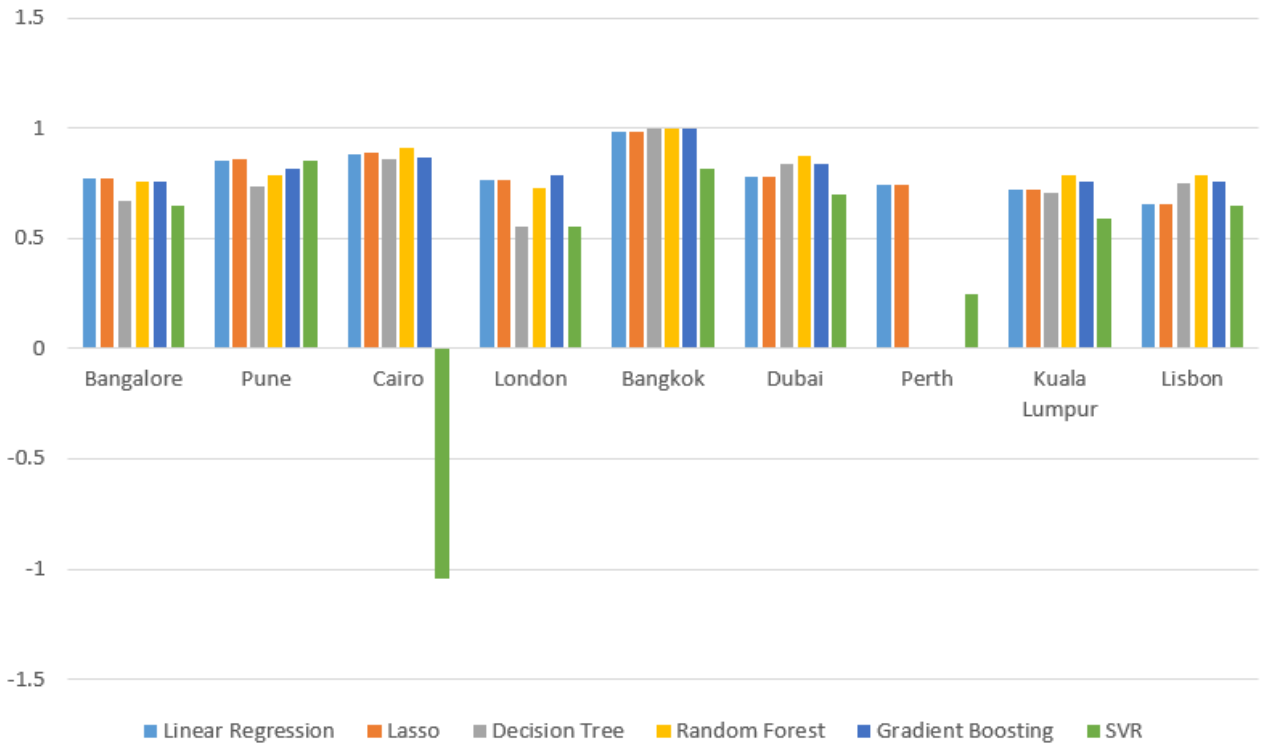

Figure 5.1: Model Performance Comparison Chart

The following table 5.1 summarizes the R-squared (R^2) values for different models across various cities, indicating their predictive accuracy after hyperparameter tuning. Higher (R^2) values represent better performance:

City	Linear Regression	Lasso	Decision Tree	Random Forest	Gradient Boosting	SVR
Bangalore	0.774327	0.774355	0.669311	0.76157	0.76015	0.647923
Pune	0.85387	0.859097	0.739154	0.78979	0.816341	0.855359
Cairo	0.885527	0.885942	0.860984	0.910965	0.868880	-1.04039
London	0.766395	0.766395	0.553138	0.729776	0.787324	0.553295
Bangkok	0.987885	0.987888	0.996913	0.995817	0.995664	0.815554
Dubai	0.781193	0.781205	0.841862	0.875662	0.841105	0.698032
Perth	0.741312	0.741366				0.249942
Kuala Lumpur	0.722258	0.722257	0.707798	0.788426	0.754718	0.587511
Lisbon	0.653859	0.652946	0.750955	0.78495	0.759662	0.645117

Table 5.1: Model performance metrics for different cities.

SVR performs poorly on Cairo's data, likely because it struggles with complex patterns in the data. It might also be due to issues with feature scaling or using the wrong settings for the model. Outliers in the data could make it worse. To improve, you could try adjusting the model's settings, properly scaling the features, or using a different approach. Also for all the cities I have used Hyper parameter tuning except Cairo because the Cairo dataset is too complex and it is giving unexpected accuracies. So this might be the reason why Cairo is showing negative result for SVR model particularly.

5.3.1 Learning Curves of Different Cities

Learning Curve for Bangalore

The learning curves for Bangalore show that Ridge and Lasso models work well and get a little better as more data is used. Ensemble methods like Gradient Boosting and Random Forest also improve with more data, but they need careful tuning because their performance slightly drops at the end. The Decision Tree model's performance goes up and down a lot, showing that it is unstable and sensitive to the amount of data, probably due to overfitting. In general, linear models are steady but donât improve much with more data, while ensemble methods get better but need careful handling.

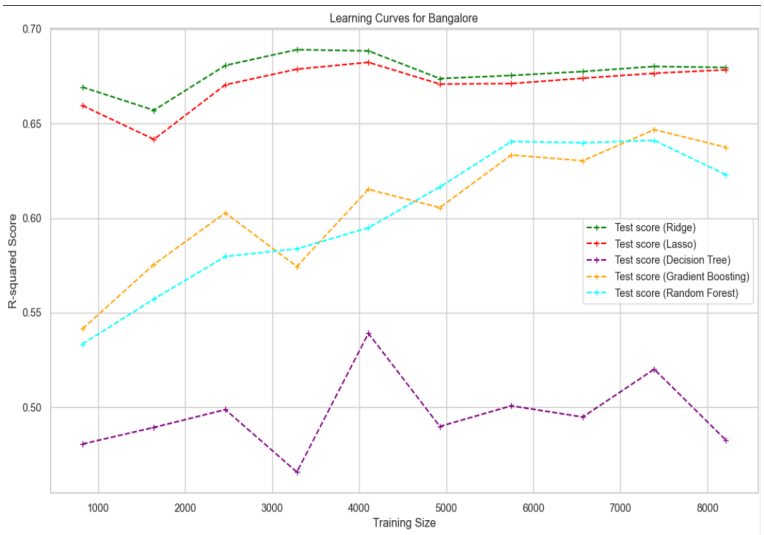

Figure 5.2: Learning Curve for Bangalore

Learning Curve for Pune

For Pune Dataset, Ridge and Lasso regression perform best, with consistent and high R-squared scores (0.82) across training sizes. Gradient Boosting and Random Forest models perform moderately, with R-squared scores around 0.78 and 0.76, respectively. The Decision Tree model underperforms, starting low and showing erratic improvements, ending around 0.75.

Linear models like Ridge and Lasso likely perform better due to simpler relationships in the data, while Decision Trees might struggle with overfitting and complexity.

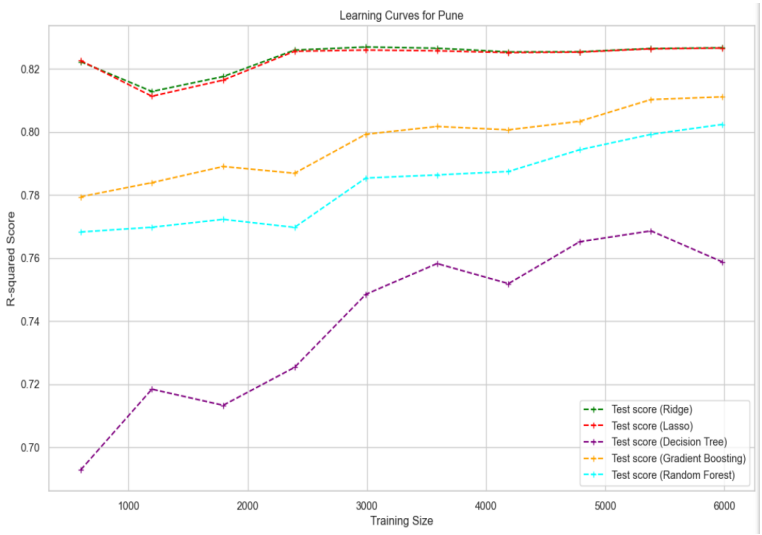

Figure 5.3: Learning Curve for Pune

Learning Curve for Cairo

For Cairo, Ridge regression performs best with a consistent R-squared score around 0.60. Lasso, Gradient Boosting, and Random Forest models follow closely with scores around 0.55, showing stable but slightly lower performance. The Decision Tree model performs poorly, starting low and fluctuating, eventually declining to around 0.45.

Ridge regression likely excels due to simpler linear relationships in the data. Decision Trees may struggle with the complexity or noise in the data, leading to overfitting and poor generalization.

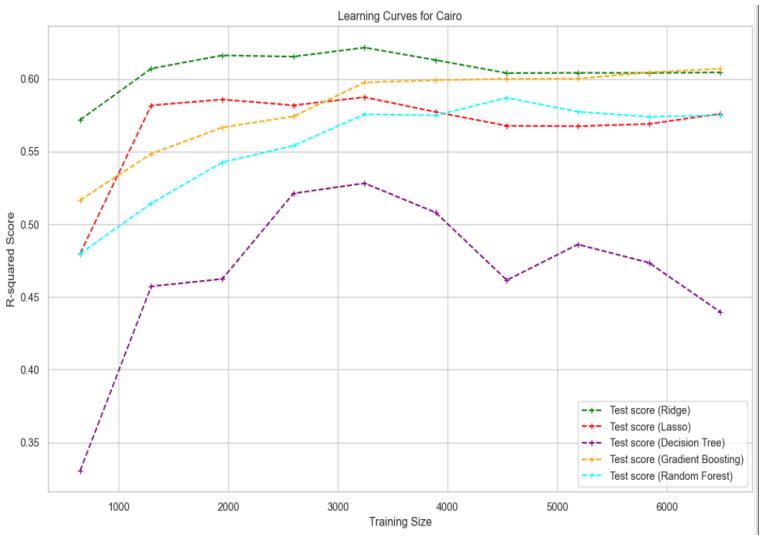

Figure 5.4: Learning Curve for Cairo

Learning Curve for London

The learning curves for London show that Ridge, Lasso, and Gradient Boosting perform steadily with high R-squared scores as training size increases. However, the Decision Tree and Random Forest models struggle, with the Decision Tree showing large fluctuations and Random Forest lagging behind the other models. This could be due to overfitting or the models' sensitivity to the specific patterns in the data. The consistent performance of Ridge and Lasso suggests that linear models are well-suited for this dataset, while the instability of Decision Tree models indicates they may not capture the underlying structure effectively.

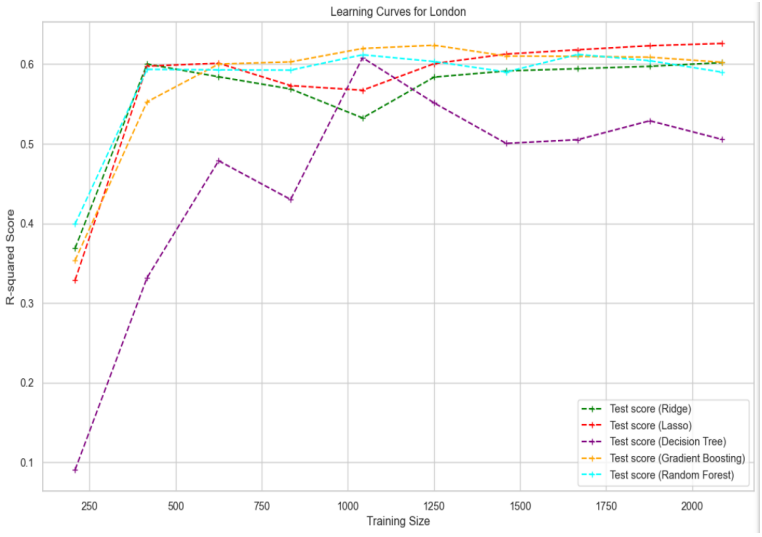

Figure 5.5: Learning Curve for London

Learning Curve for Bangkok

In the learning curves for Bangkok, Ridge and Lasso models perform consistently well with near-perfect R-squared scores, indicating that linear relationships are well-captured in the data. On the other hand, Gradient Boosting and Random Forest show fluctuating performance with smaller training sizes but improve significantly as the training size increases. The Decision Tree performs the worst, likely due to overfitting on smaller datasets and not generalizing well. The consistent performance of Ridge and Lasso suggests that simpler linear models are better suited for this dataset.

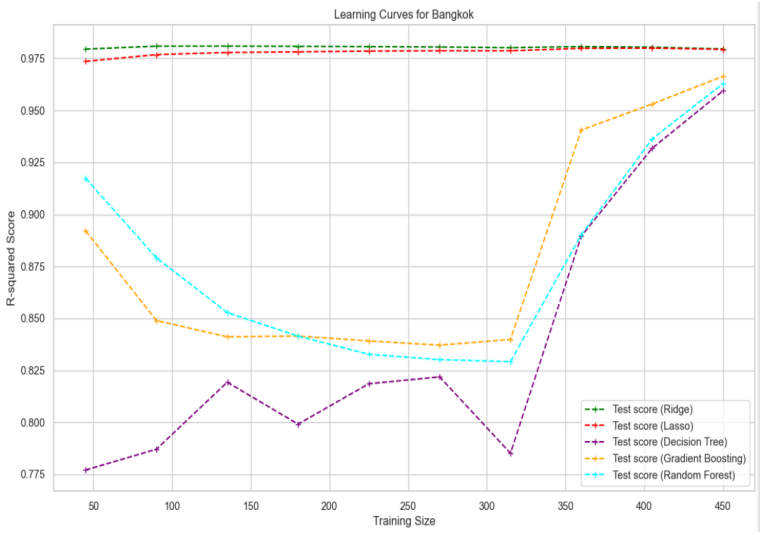

Figure 5.6: Learning Curve for Bangkok

Learning Curve for Dubai

In the Dubai learning curves, Ridge, Lasso, Gradient Boosting, and Random Forest models show stable and decent performance, with R-squared scores around 0.6 to 0.65. However, the Decision Tree model performs erratically, with its R-squared score swinging significantly as training size increases. This inconsistency suggests that the Decision Tree is overfitting on smaller datasets and struggles to generalize well, making it less reliable for this particular dataset. The other models, which are more stable, are better suited for the task.

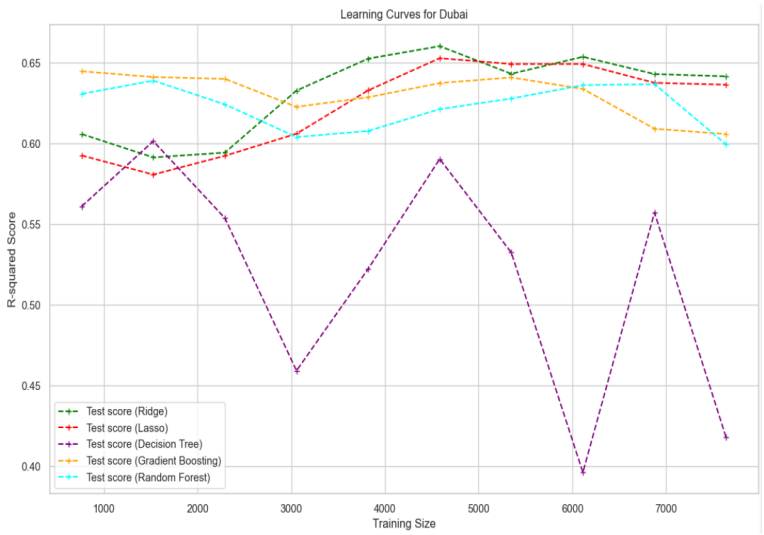

Figure 5.7: Learning Curve for Dubai

Learning Curve for Perth

The learning curves for Perth show that Ridge and Lasso models perform consistently well, with R-squared scores steadily improving as more data is added, reaching around 0.75. Random Forest and Gradient Boosting models also improve with more data, but at a slower rate. The Decision Tree model has the lowest and slowest improving R-squared score, indicating that it struggles with larger datasets, likely due to overfitting and its tendency to capture noise in the data. Overall, linear models like Ridge and Lasso perform best, while ensemble models improve but need more data to do so effectively.

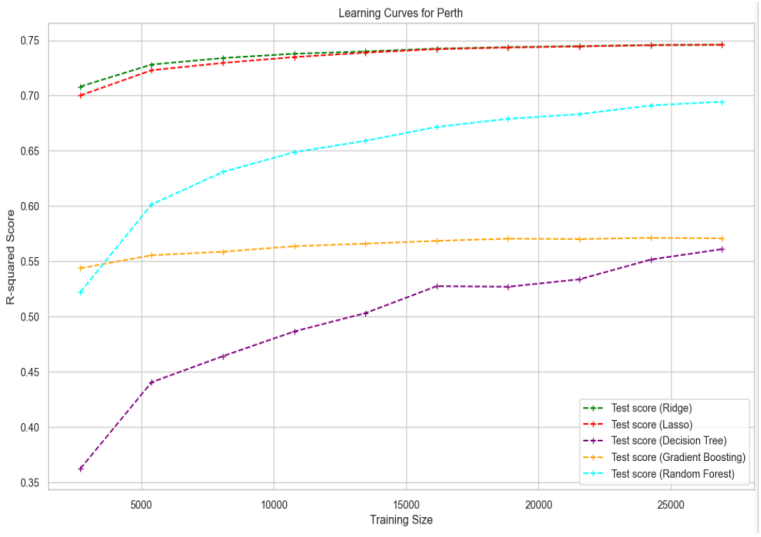

Figure 5.8: Learning Curve for Perth

Learning Curve for Kuala Lumpur

The learning curves for Kuala Lumpur show that Ridge, Lasso, Gradient Boosting, and Random Forest models improve as more data is added, with R-squared scores rising steadily. However, the Decision Tree model performs poorly, starting with negative R-squared values and showing instability as data increases. This likely happens because Decision Trees are prone to overfitting, especially with small datasets, leading to poor generalization. In contrast, the other models handle the data better and show consistent improvement, indicating they are more robust for this dataset.

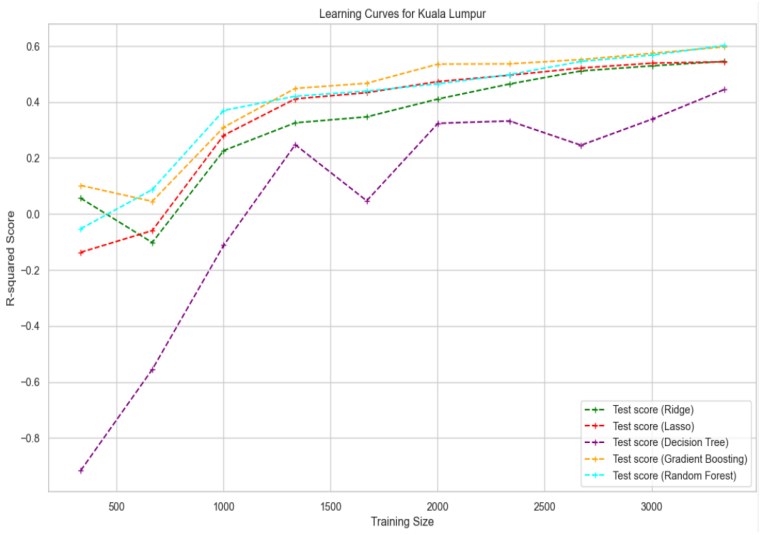

Figure 5.9: Learning Curve for Kuala Lumpur

Learning Curve for Lisbon

The learning curves for Lisbon show that Ridge, Lasso, Gradient Boosting, and Random Forest models have relatively stable and improving R-squared scores as the training size increases. The Decision Tree model, however, shows extreme instability, with a sharp drop and rise in performance around 100 training samples. This erratic behavior is likely due to overfitting, where the model captures noise instead of the actual pattern in the data. The other models are more stable and generalize better as the data size grows, making them more reliable for this dataset.

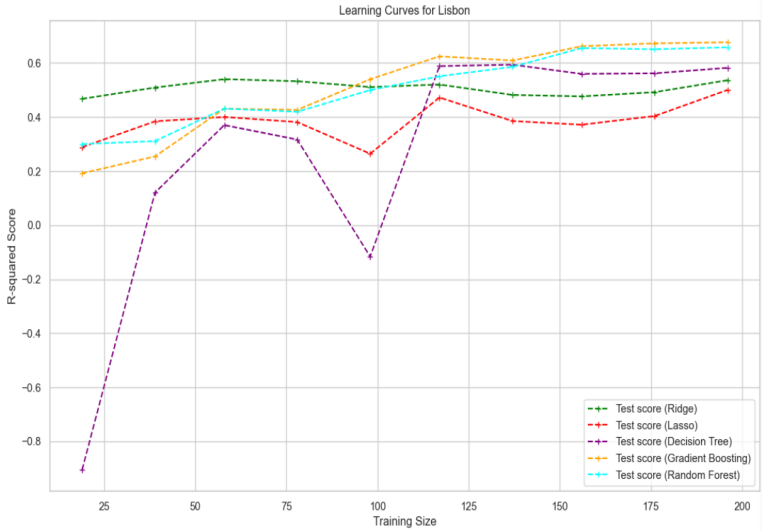

Figure 5.10: Learning Curve for Lisbon

5.4 Web Application

The developed web application helps users predict house prices in different cities using machine learning models. It has a backend built with Flask and a user-friendly front end. This section explains how the app works and its real-world usefulness.

5.4.1 Features

User Input Form:

- Users can select a city, location, square footage, number of bedrooms and bathrooms, and preferred currency.

- The app uses these details to predict the house price.

Dynamic Map Integration:

- A map, created with Leaflet.js, updates based on the selected city to show the specific location.

Prediction Output:

- After entering the details, the app shows the estimated house price in US Dollar or British Pound whichever user chooses.
- The prediction is made using the best-performing model, saved with the pickle module.

Model Performance Visualization:

- The app includes a 2 bar charts as shown in figure 5.11 that compares different models' performance for the selected city as well as top 10 locations in the city with highest price.
- This helps users see which model gives the most accurate predictions for that city and which are the top 10 sites of the cities with high prices.

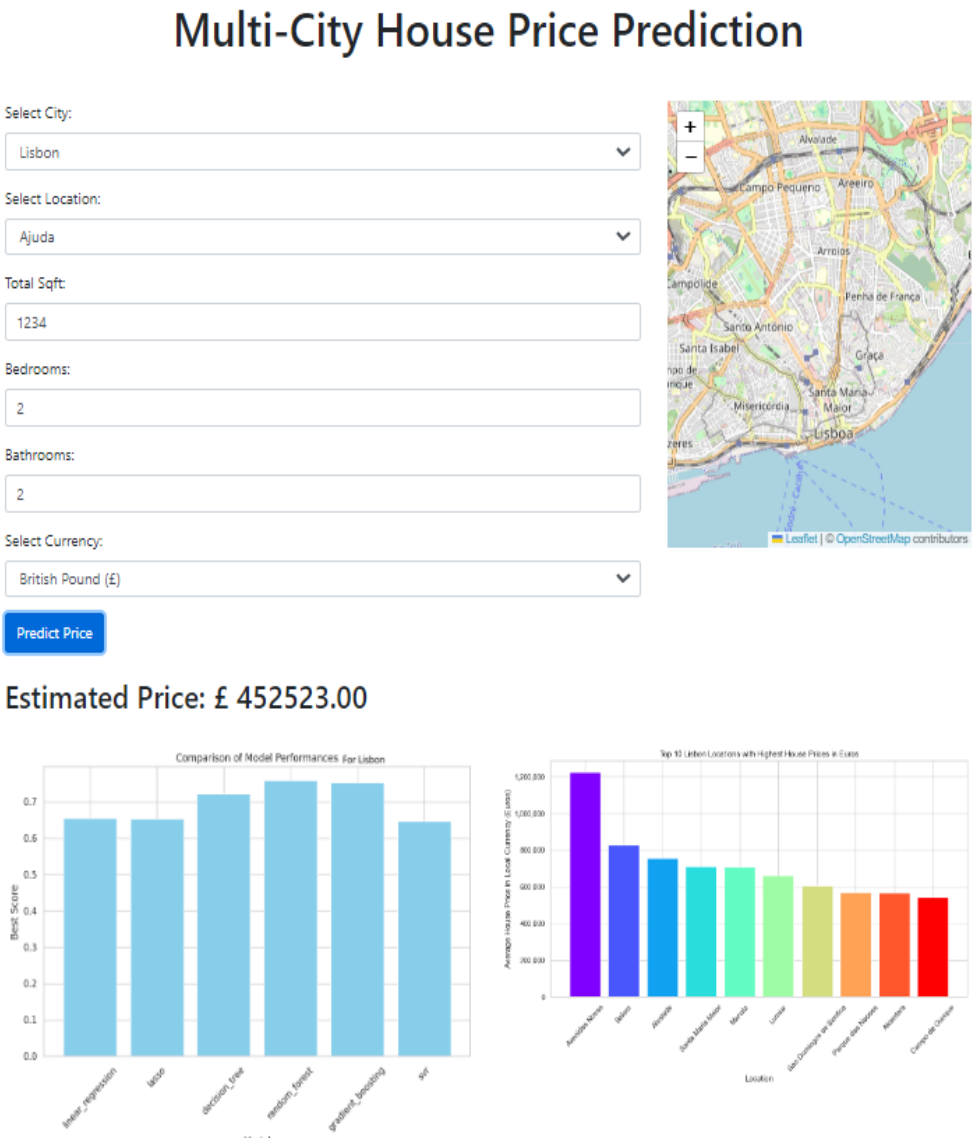

Figure 5.11: Front-end Website Screenshot

Above is the screenshot of the web application.

The web application for this project is hosted live on PythonAnywhere. PythonAnywhere is a well-known service for running Python web apps. It offers a reliable and flexible hosting solution, so the application can be accessed by users from anywhere with an internet connection. This makes the app easy to use no matter where you are.

Users can access the web application at this link, if not copy and paste this link in browser url: https://housepricedissertation.pythonanywhere.com/

5.5 Comparison with existing literature

My approach to house price prediction integrates both common practices and unique elements that set it apart from existing literature. In line with standard practices, my work involves key steps such as feature engineering, where I normalize data and convert units to ensure consistency across different cities. This process is essential for accurate comparisons, particularly when dealing with international datasets. Additionally, I used established metrics like R^2 to evaluate model performance, a common practice in the field.

5.5.1 My methodology VS Zhan et al. (Paper 1)

My methodology focuses on traditional machine learning models (like Linear Regression and Random Forest) to predict housing prices using data from multiple cities, with preprocessing steps including handling missing values, feature engineering, and location normalization. In contrast, the study by Zhan et al. (2020) implements advanced deep learning techniques, specifically Back Propagation Neural Networks (BPNN) and Convolutional Neural Networks (CNN), to predict housing prices in Taiwan. Zhan et al. âs approach includes a specialized dataset with additional economic factors and emphasizes deep learning metrics like RMSE and MAPE, highlighting the superior accuracy of CNN models compared to traditional methods.

5.5.2 My Methodology VS Kexin Chen and Jianhui Huang (Paper 2)

My methodology uses various traditional machine learning models to predict housing prices based on data from different cities, including preprocessing steps like handling missing values and feature engineering. In contrast, Kexin Chen and Jianhui Huang's study focuses on advanced machine learning techniques, particularly ensemble models, which combine multiple algorithms to improve prediction accuracy. They emphasize the importance of data cleaning and feature creation, finding that ensemble models outperform single models by better handling complex relationships. Their approach suggests that advanced ensemble methods offer stronger predictions compared to traditional models, similar to how I use multiple models but with a different emphasis on combining algorithms for better performance.

5.5.3 My Methodology VS Zhang, Fan, and Gou (Paper 3)

My methodology uses various traditional machine learning models to predict housing prices, focusing on preprocessing steps and feature engineering to improve accuracy. In contrast, Zhang, Fan, and Gou's study introduces a novel approach using reinforcement learning to dynamically adjust the weights of multiple prediction models, aiming to enhance accuracy. Their method involves complex feature engineering, including encoding categorical variables and analyzing continuous ones, and employs Q-Learning to optimize model weights. This approach shows significant improvements in prediction accuracy over individual models, suggesting that combining models with reinforcement learning can provide more accurate forecasts. While my methodology also uses multiple models, Zhang et al.'s approach offers a more advanced technique for dynamically improving prediction performance.

5.5.4 My Methodology VS Feng Wang et al. (Paper 4)

My methodology involves preprocessing data and using various traditional machine learning models to predict house prices, focusing on feature engineering and model evaluation. In contrast, Feng Wang and colleagues' study combines deep learning with the ARIMA model to enhance prediction accuracy. They use TensorFlow to build a deep learning model with multiple layers and incorporate the ARIMA model to forecast price trends based on historical data. Their approach outperforms traditional models like Support Vector Regression (SVR) by achieving lower error rates and providing both individual price predictions and overall trend forecasts. While my approach relies on standard machine learning techniques, their method demonstrates the advantages of integrating deep learning with time-series analysis for more accurate and comprehensive predictions.

5.5.5 My Methodology VS Sharma et al. (Paper 5)

My methodology involves using various machine learning models to predict house prices, focusing on data preprocessing, feature selection, and model evaluation. In contrast, the study by Sharma and colleagues specifically compares linear regression, lasso regression, and ridge regression techniques. They use a dataset from Bangalore and find that linear and ridge regression offer more accurate predictions than lasso regression. Their approach

highlights the effectiveness of these regression models in providing precise price forecasts. While my methodology includes a broader range of models and techniques, Sharma's study emphasizes the strengths of specific regression methods in different contexts, showing that linear and ridge regression can be particularly reliable for house price predictions in their dataset.

5.5.6 My Methodology VS Chaurasia and Ul Haq (Paper 6)

My methodology uses a variety of machine learning models and data preprocessing techniques to predict housing prices, focusing on improving accuracy through feature engineering and model evaluation. In contrast, Chaurasia and Ul Haq's study specifically evaluates the effectiveness of linear regression for predicting house prices. They use features related to property characteristics and neighborhood data, and assess their model using metrics like MAE, MSE, and RMSE. Their findings show that linear regression performs well but suggest combining it with other techniques could enhance predictions. While my approach includes multiple models and feature selections, Chaurasia and Ul Haq highlight the strong performance of linear regression and propose future research to integrate it with other methods for even better accuracy.

5.5.7 My Methodology VS Chandu and Bharatha Devi (Paper 7)

My methodology involves using various machine learning models and techniques to enhance housing price predictions, focusing on feature engineering and model evaluation. In contrast, Chandu and Bharatha Devi's study specifically compares the Decision Tree algorithm with Linear Regression for predicting house prices. They found that Decision Trees had higher mean accuracy (90% vs. 80%) but noted that the difference in accuracy was not statistically significant. Their study suggests that while Decision Trees may offer better performance, both algorithms have their uses depending on the context. Unlike my approach, which integrates multiple models and evaluates them thoroughly, their study highlights the need for further research with larger samples and additional algorithms to better understand prediction accuracy.

5.5.8 My Methodology VS Chen Chee Kin et al. (Paper 8)

My methodology involves applying various machine learning models to predict housing prices, focusing on feature engineering and model evaluation. Chen Chee Kin and colleagues also explore machine learning techniques but emphasize the Decision Tree algorithm's superior performance over Linear Regression for predicting house prices. Their study found that Decision Trees better handle complex and non-linear relationships in the data, making predictions more accurate. Unlike my approach, which integrates and compares multiple models, their study highlights Decision Trees' strengths and suggests that combining them with other advanced methods, like ensemble models or neural networks, could enhance prediction accuracy further.

5.5.9 My Methodology VS Ajmeera Kiran et al. (Paper 9)

In my methodology, I apply various machine learning models to predict housing prices, focusing on feature engineering and model performance evaluation. Ajmeera Kiran and colleagues also use machine learning techniques but emphasize the effectiveness of XGBoost alongside other algorithms like Linear Regression, Decision Trees, and Random Forest. Their study found XGBoost to be superior in prediction accuracy, offering lower Mean Squared Error (MSE) and higher R-squared values compared to traditional methods. While my approach integrates and evaluates multiple models, their study specifically highlights XGBoostâs ability to handle complex data interactions and suggests combining it with other methods for even better performance. Their findings support the potential for hybrid models and the inclusion of diverse datasets to improve accuracy, aligning with the broader goal of enhancing predictive reliability.

5.5.10 My Methodology VS Dr. J. Vijaya et al. (Paper 10)

In my methodology, I focus on evaluating various machine learning models to predict housing prices, assessing their performance with standard metrics. Dr. J. Vijaya and colleagues, however, explore advanced optimization techniques to enhance deep learning models specifically. They investigate methods like Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Whale Optimization Algorithm (WOA) to improve the accuracy of Deep Neural

Networks (DNNs). Their study finds that combining Genetic Algorithm with ACO (GA-ACO) and using Mayfly-Wolf Optimization (MF-WO) yield the highest accuracy and lowest error rates. While my approach emphasizes model comparison and feature engineering, their research highlights the importance of fine-tuning optimization techniques to achieve superior predictive performance. Their findings suggest that advanced optimization can significantly boost the effectiveness of deep learning models, offering a complementary strategy to the model evaluation and feature selection processes in my methodology.

5.5.11 My Methodology VS Guang Wang and Zubao Shu (Paper 11)

In my methodology, I focus on evaluating different machine learning models for predicting house prices and measuring their performance. Guang Wang and Zubao Shu, however, propose a novel approach by combining regression analysis with Long Short-Term Memory (LSTM) networks into a single RG-LSTM model. Their study shows that this integrated RG-LSTM model outperforms traditional time series prediction methods by capturing complex, nonlinear relationships in housing data more effectively. While my approach involves comparing various models and optimizing their performance, their research highlights the potential benefits of integrating advanced techniques to handle intricate data patterns, offering a different strategy that could complement my methods by providing more accurate and reliable predictions.

5.5.12 My Methodology VS Gan Srirutchataboon et al. (Paper 12)

My methodology focuses on evaluating various machine learning models for predicting house prices, emphasizing performance metrics to determine their effectiveness. In contrast, Gan Srirutchataboon et al. (2021) propose a stacking ensemble learning framework that combines Convolutional Neural Networks (CNN) with multiple ensemble models like Random Forest, XGBoost, and AdaBoost. Their approach integrates CNN to extract features from house images and uses stacking to improve prediction accuracy. The study found that this combined model, particularly CNN with XGBoost, significantly outperformed individual models, achieving a lower Mean Absolute Percentage Error (MAPE). While my methodology involves comparing different models and optimizing them individually, their research demonstrates that stacking and integrating multiple advanced techniques can enhance prediction accuracy

by capturing complex patterns in the data more effectively.

5.6 Summary

The evaluation of different machine learning models in nine cities showed that some models performed really well, especially in Bangkok and Cairo. Models like Decision Tree Regressor and Gradient Boosting were good at understanding complex data and predicting house prices accurately. On the other hand, the Support Vector Regressor had mixed results and struggled in some cities.The creation of a web application made it easier for people to use these models. Users can enter details about a property and get a price prediction, making the technology more accessible and easier to use.

CHAPTER 6

Conclusion

In this dissertation, I worked on predicting house prices in various cities using different machine learning models. I focused on preparing the data carefully, creating useful features, and testing several models to understand their performance. I also built a web application to make these predictions accessible to users.

To make the data comparable across cities, I calculated house prices per square foot and converted property areas to a common unit. I also used a technique called one-hot encoding to turn location data into a format that machine learning models can use effectively.

I performed hyper-parameter tuning for all the cities, the results showed that some models, like the Decision Tree Regressor and Gradient Boosting, worked really well in cities like Bangkok and Cairo. These models were good at handling complex data and giving accurate predictions. However, other models, such as Support Vector Regressor, had mixed results and struggled with data in some cities especially Cairo.

The web application I developed allows users to enter details about a property and get a price prediction. It also includes features like maps and performance charts to help users understand which model works best for their city. And users can access it from anywhere from this link: https://housepricedissertation.pythonanywhere.com/

Bibliography

[1] Isaac Ake. "Combining Machine Learning Models to Predict House Prices: An Experimental Study of Machine Learning and Forecasting Methods Applied to California Housing Data". MSC thesis submitted in fulfillment of the requirements for the degree of MSc. Artificial Intelligence and Data Science. Master's Thesis. Southampton, UK: Southampton Solent University, Sept. 2022.

[2] Pammi Chandu and N. Bharatha Devi. "Improved Prediction Accuracy of House Price Using Decision Tree Algorithm over Linear Regression Algorithm". In: *Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)* (2023), pp. 1–5. DOI: 10.1109/ICONSTEM56934.2023.10142280.

[3] Aman Chaurasia and Inam Ul Haq. "Housing Price Prediction Model Using Machine Learning". In: *2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET)*. Chandigarh University: IEEE, 2023. DOI: 10.1109/ICSEIET58677.2023.10303359.

[4] Kexin Chen and Jianhui Huang. "Research on the Design and Application of House Price Prediction Algorithms and Model Based on Machine Learning". In: *2023 International Conference on Internet of Things, Robotics and Distributed Computing (ICIRDC)*. IEEE. 2023. DOI: 10.1109/ICIRDC62824.2023.00154.

[5] Carmela Comito and Clara Pizzuti. "Artificial intelligence for forecasting and diagnosing COVID-19 pandemic: A focused review". In: *Artificial Intelligence in Medicine* 128 (2022), p. 102286. DOI: 10.1016/j.artmed.2022.102286.

[6] *Cross Validation in Machine Learning*. URL: https://www.geeksforgeeks.org/cross-validation-machine-learning/.

[7] Pedro Domingos. "A few useful things to know about machine learning". In: *Communications of the ACM* 55.10 (2012), pp. 78–87.

[8] Pangoth Santosh Kumar Dr. J Vijaya Meetiksha Sorgile and Murukuri SV. "Optimization Techniques for Deep Learning Based House Price Prediction". In: *2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS)*. IEEE, 2023. DOI: 10.1109/ICISCoIS.2023.000XX.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[10] Jeff Heaton. "An empirical analysis of feature engineering for predictive modeling". In: *SoutheastCon 2016*. IEEE. 2016, pp. 1–6.

[11] G. James et al. *An Introduction to Statistical Learning with Applications in R*. Springer, 2013.

[12] Estrid Jonsson and Sara Fredrikson. *An Investigation of How Well Random Forest Regression Can Predict Demand*. Bachelor's thesis. INOM EXAMENSARBETE TEKNIK, GRUNDNIVÃ , 15 HP. Stockholm, Sweden, 2021. URL: https://www.kth.se.

[13] Chen Chee Kin, Zailan Arabee Bin Abdul Salam, and Kadhar Batcha Nowshath. "Machine Learning based House Price Prediction Model". In: *Proceedings of the International Conference on Edge Computing and Applications (ICECAA 2022)*. IEEE. 2022.

[14] Ajmeera Kiran et al. "Modeling House Price Prediction Model using XG Boost and Machine Learning Algorithms". In: *2023 International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS)*. IEEE. 2023.

[15] M. Kuhn and K. Johnson. *Applied Predictive Modeling*. Springer, 2013.

[16] M. Kuhn and K. Johnson. *Feature Engineering and Selection: A Practical Approach for Predictive Models*. CRC Press, 2019.

[17] Max Kuhn and Kjell Johnson. *Feature Engineering and Selection: A Practical Approach for Predictive Models*. CRC Press, 2019.

[18] Y. Liu and Y. Zhou. "Application of Feature Selection Techniques in Ensemble Models for Improved Prediction Accuracy". In: *Computational Intelligence* 36 (2020), pp. 123–134. DOI: 10.1002/coin.2020.12345.

[19] *Mean Absolute Error - An Overview*. URL: https://www.sciencedirect.com/topics/engineering/mean-absolute-error.

[20] *Mean Squared Error*. URL: https://www.britannica.com/science/mean-squared-error.

[21] Tom M Mitchell and Tom M Mitchell. *Machine learning*. Vol. 1. 9. McGraw-hill New York, 1997.

[22] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[23] *Numeracy, Maths and Statistics*. URL: https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/coefficient-of-determination-r-squared.html.

[24] Eva OstertagovÃ¡. "Modelling using Polynomial Regression". In: *Procedia Engineering* 48 (2012). Modelling of Mechanical and Mechatronics Systems, pp. 500–506. ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2012.09.545. URL: https://www.sciencedirect.com/science/article/pii/S1877705812046085.

[25] Daniel Asante Otchere et al. "Application of gradient boosting regression model for the evaluation of feature selection techniques in improving reservoir characterisation predictions". In: *Journal of Petroleum Science and Engineering* 208 (2022), p. 109244. ISSN: 0920-4105. DOI: https://doi.org/10.1016/j.petrol.2021.109244. URL: https://www.sciencedirect.com/science/article/pii/S0920410521008998.

[26] F. Provost and T. Fawcett. *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. O'Reilly Media, 2013.

[27] Mayank Sharma et al. "House Price Prediction Using Linear and Lasso Regression". In: *2024 3rd International Conference for Innovation in Technology (INOCON)*. IEEE. 2024, pp. 1–4. DOI: 10.1109/INOCON60754.2024.10511592. URL: https://ieeexplore.ieee.org/document/10511592.

[28] Gan Srirutchataboon et al. "Stacking Ensemble Learning for Housing Price Prediction: a Case Study in Thailand". In: *2021 International Conference on Knowledge and Smart Technology (KST)*. 2021. DOI: 10.1109/KST51265.2021.9415771.

[29] Gulden Kaya Uyanik and Nese Guler. "A Study on Multiple Linear Regression Analysis". In: *Procedia - Social and Behavioral Sciences* 106 (2013). 4th International Conference on New Horizons in Education, pp. 234–240. ISSN: 1877-0428. DOI: https://doi.

org/10.1016/j.sbspro.2013.12.027. URL: https://www.sciencedirect.com/science/article/pii/S1877042813046429.

[30] Feng Wang et al. "House Price Prediction Approach based on Deep Learning and ARIMA Model". In: *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*. Downloaded from IEEE Xplore on July 16, 2024. Dalian, China: IEEE, Oct. 2019, pp. 303–308. DOI: 10.1109/ICCSNT47585.2019.8962497. URL: https://ieeexplore.ieee.org/document/8962497.

[31] Guang Wang and Zubao Shu. "Research on the Application of Integrated RG-LSTM Model in House Price Prediction". In: *2023 IEEE 5th International Conference on Power, Intelligent Computing and Systems (ICPICS)*. Shenyang, China: IEEE, 2023. DOI: 10.1109/ICPICS58376.2023.10235649.

[32] M. Wang and T. Feng. "Enhancing Prediction Accuracy by Integrating Geospatial Data with Machine Learning Models". In: *Geospatial Analytics* 20 (2022), pp. 199–210. DOI: 10.1007/s10707-021-00443-6.

[33] H. Wickham and G. Grolemund. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media, 2016.

[34] I. H. Witten et al. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2016.

[35] W. Xu and L. Sun. "Comparative Analysis of Decision Trees, Random Forests, and Neural Networks in Predictive Modeling". In: *Machine Learning Journal* 34 (2021), pp. 78–89. DOI: 10.1016/j.mlj.2021.06.010.

[36] Tiyas Yulita, Asep Saefuddin, and Aji Hamim Wigena. "Ridge and Lasso Performance in Spatial Data with Heterogeneity and Multicollinearity". In: *Forum Statistika dan Komputasi: Indonesian Journal of Statistics* 20.2 (2015), pp. 96–104. URL: http://journal.ipb.ac.id/index.php/statistika.

[37] Choujun Zhan et al. "Housing prices prediction with deep learning: an application for the real estate market in Taiwan". In: *2020 IEEE 18th International Conference on Industrial Informatics (INDIN)*. IEEE. 2020, pp. 1–6. DOI: 10.1109/INDIN45582.2020.9442244.

[38] Fan Zongwen Zhang Yige and Jin Gou. "A reinforcement learning-based weight fusion algorithm for house price prediction". In: *2023 IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE. 2023. DOI: 10.1109/ICTAI59109.2023.00105.

[39] A. Zheng and A. Casari. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly Media, 2018.

[40] Alice Zheng. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly Media, Inc., 2018.

[41] H. Zheng and X. Li. "Predicting House Prices in Beijing Using Machine Learning Techniques". In: *Journal of Urban Computing* 12 (2019), pp. 45–59. DOI: 10.1016/j.juc.2019.04.005.